

# JST-CREST

## 研究領域

「実用化を目指した組込みシステム用ディペンダブル・オペレーティングシステム」

## DEOS プロジェクト



## DS-Bench/Test-Env(D-Cloud)

## 環境構築手順書

Version 1.00

2013/05/01

DEOS 研究開発センター

## 目次

---

1	はじめに.....	4
2	環境構成 .....	5
2.1	仮想環境を使用しない構成.....	5
2.2	仮想環境を使用する場合.....	6
3	D-Cloud .....	8
3.1	Ubuntu12.04 への対応.....	8
3.2	OpenStack のインストール手順.....	8
3.3	D-Cloud コントローラのインストール手順 .....	8
3.3.1	OpenStack をインストールした場合 .....	8
3.3.1.1	deb パッケージからインストールする場合 .....	8
3.3.2	OpenStack をインストールしていない場合 .....	9
3.3.2.1	deb パッケージからインストールする場合 .....	9
4	OpenStack .....	10
4.1	OpenStack コントローラ+計算ノード構築手順 .....	10
4.1.1	ネットワークの設定 .....	10
4.1.2	ホスト名の追加.....	10
4.1.3	リポジトリの追加.....	11
4.1.4	OpenStack のインストール .....	11
4.1.5	D-Cloud 用のプロジェクト作成 .....	17
4.2	OpenStack 計算ノード構築手順.....	19
4.2.1	ホスト名の追加.....	19
4.2.2	リポジトリの追加.....	19
4.2.3	OpenStack(計算ノード)のインストール .....	19
5	DS-Bench.....	21
5.1	Ubuntu12.04 への対応.....	21
5.2	コントローラのインストール手順 .....	21
5.2.1	CPAN シェルの設定.....	21
5.2.2	DS-Bench コントローラのインストール.....	21
5.2.3	Apache の設定 .....	22
5.2.4	Sedna DB サーバの設定(任意) .....	23
5.2.5	D-Cloud コントローラの設定(任意).....	23
5.3	SednaDB サーバのインストール.....	24
5.3.1	deb パッケージからインストール .....	24
5.3.2	Sedna の修正 .....	24

5.4	DS-Bench ターゲットのインストール .....	25
5.4.1	deb パッケージからインストール.....	25
5.4.2	リポジトリファイルの修正 .....	25
6	仮想マシン.....	26
6.1	仮想ターゲット使用時のネットワーク設定 .....	26
6.2	仮想ターゲットイメージ作成手順.....	27
6.2.1	初期イメージの登録.....	27
6.2.2	初期イメージの起動.....	28
6.2.3	仮想ターゲットイメージの作成.....	29
6.2.4	仮想ターゲットマシンの登録.....	33

## 1 はじめに

この DS-Bench/Test-Env (D-Cloud) 環境構築手順書（以下、本書）では、ディペンダビリティの計測ツールである DS-Bench、および資源を管理してシステムテストを迅速に行うツールである Test-Env (D-Cloud) のインストール、設定手順、また仮想マシンの作成手順について説明します。

クラウドとして計算資源を適切に管理するために用いる OpenStack については、Test-Env (D-Cloud) コントローラが起動しているサーバにインストール及び各種設定を行ってください。物理マシンのみを使用する場合は、この手順および、仮想マシン作成手順を実施する必要はありません。

使用方法については別途、DS-Bench/Test-Env 実行手順書 (DEOS-FY2013-BT-01J) を参照してください。

**※資源を管理してシステムテストを迅速に行うツールである Test-Env について、本書では D-Cloud と表記します。**

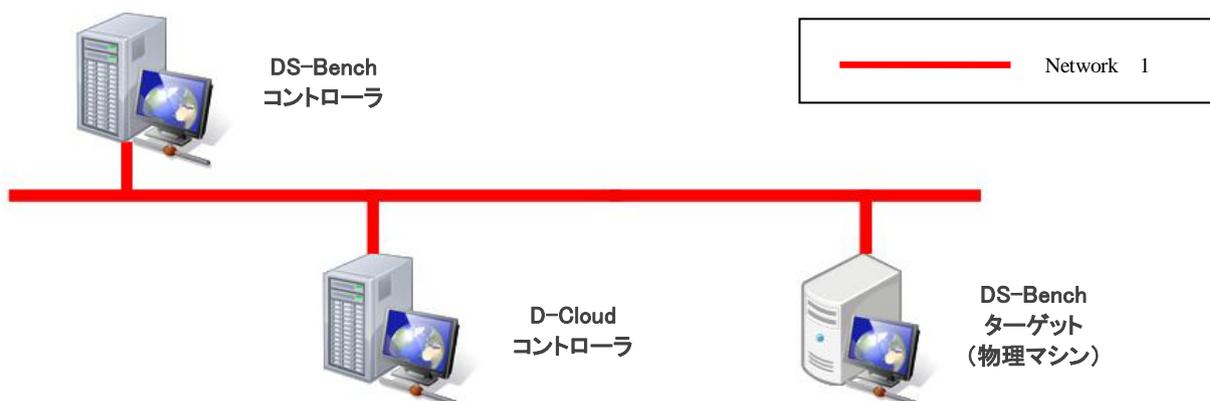
**また、本書に記載されているシステム名、製品名、サービス名などは一般に各社の商標または登録商標です。**

## 2 環境構成

本書では以下の構成でインストール、設定を行います。

### 2.1 仮想環境を使用しない構成

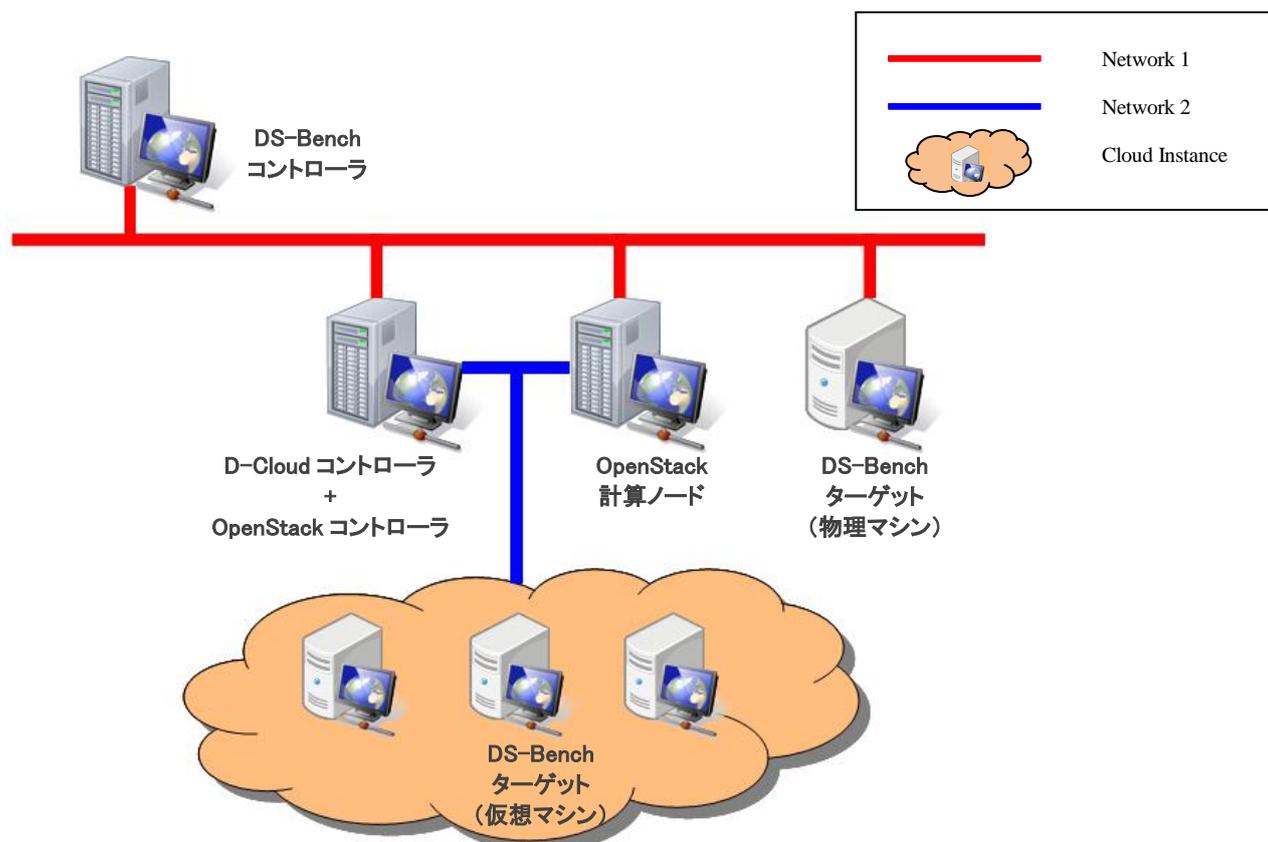
- DS-Bench コントローラ  
Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1  
Sedna XML DB
- DS-Bench ターゲット (物理マシン)  
Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1
- D-Cloud コントローラ  
Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1



ノード	概要	詳細
D-Cloud コントローラ	計算資源(DS-Bench ターゲット)などの情報を一元管理するために、計算資源の情報取得、予約、起動等を D-Cloud 経由で行い、また Fault Injection の管理と実行を行います。仮想環境を使用しないので D-Cloud の <code>withoutOpenStack.deb</code> をインストールします。	D-Cloud コントローラ <a href="#">3.1</a> <a href="#">3.3</a>
DS-Bench コントローラ	ベンチマークツール、シナリオデータを持ち、ターゲットに対してシナリオ実行命令を送り、その結果を取得、保存する機能を有します。また、WebUI を持ち、これらの機能を直観的、容易に操作できます。DS-Bench コントローラとデータベースとして Sedna を使用します。	DS-Bench コントローラ <a href="#">5.1</a> <a href="#">5.2</a> Sedna DB <a href="#">5.3</a>
DS-Bench ターゲット (物理マシン)	DS-Bench コントローラからの要求を受け、テスト実行、異常生成を行い、その結果を成型して返却します。DS-Bench ターゲット用のパッケージをインストールします。	DS-Bench ターゲット <a href="#">5.4</a>

## 2.2 仮想環境を使用する場合

- DS-Bench コントローラ  
Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1  
Sedna XML DB
- DS-Bench ターゲット (物理マシン)  
Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1
- DS-Bench ターゲット (仮想マシン)  
ubuntu-10.04-server-cloudimg-amd64 または ubuntu-12.04-server-cloudimg-amd64
- D-Cloud コントローラ + OpenStack コントローラ  
Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1  
OpenStack 2011.3 (diablo) または OpenStack 2012.1.3 (Essex)
- 計算ノード  
Ubuntu Server Edition 64-bit 10.04.3 または Ubuntu Server Edition 64-bit 12.04.1  
OpenStack 2011.3 (diablo) または OpenStack 2012.1.3 (Essex)



ノード	概要	詳細
D-Cloud コントローラ + OpenStack コントローラ	計算資源 (DS-Bench ターゲット) などの情報を一元管理するために、計算資源の情報取得、予約、起動等を D-Cloud 経由で行い、また Fault Injection の管理と実行を行います。仮想環境を構築するため D-Cloud の withOpenStack.deb と OpenStack のインストール、および設定を行います。	D-Cloud コントローラ <a href="#">3.1</a> <a href="#">3.3</a> OpenStack <a href="#">4.1</a>
DS-Bench コントローラ	ベンチマークツール、シナリオデータを持ち、ターゲットに対してシナリオ実行命令を送り、その結果を取得、保存します。また、WebUI を有し、これらの機能を直観的、容易に操作できます。DS-Bench コントローラとデータベースとして Sedna をインストールします。	DS-Bench コントローラ <a href="#">5.1</a> <a href="#">5.2</a> Sedna DB <a href="#">5.3</a>
DS-Bench ターゲット (物理マシン)	DS-Bench コントローラからの要求を受け、テスト実行、異常生成を行い、その結果を成型して返却します。DS-Bench ターゲット用のパッケージをインストールします。	DS-Bench ターゲット <a href="#">5.4</a>
OpenStack 計算ノード	D-Cloud が使用する OpenStack は通常は D-Cloud コントローラ上のみで動作します。計算ノードは、D-Cloud 上の計算資源不足した場合など必要に応じてインストール、設定を行ってください。計算ノードを追加すると計算ノードの計算資源を利用して新規に仮想マシンを作成できます。	OpenStack <a href="#">4.2</a>
DS-Bench ターゲット (仮想マシン)	DS-Bench 仮想ターゲットは、OpenStack がインストールされた D-Cloud コントローラ、または OpenStack 計算ノードで設定します。DS-Bench コントローラからの要求を受け、テスト実行、異常生成を行い、その結果を成型して返却します。	DS-Bench ターゲット <a href="#">5.4</a> 仮想マシン <a href="#">6</a>

## 3 D-Cloud

### 3.1 Ubuntu12.04 への対応

Ubuntu12.04 に DS-Bench/D-Cloud のパッケージをインストールする場合、**libdigest-sha1-perl** パッケージのインストールを行う必要があります。

以下の手順でインストールを行ってください。

```
$ wget http://launchpadlibrarian.net/85191944/libdigest-sha1-perl_2.13-2bui1ud2_amd64.deb
$ sudo dpkg -i libdigest-sha1-perl_2.13-2bui1ud2_amd64.deb
```

この後の作業は Ubuntu12.04、Ubuntu10.04 とともに同じになります。

### 3.2 OpenStack のインストール手順

仮想マシンを使用する場合は OpenStack をインストールしてください。インストール手順については、「[4 OpenStack](#)」を参照してください

物理マシンのみを使用する場合は、本項は実施する必要はありません。

### 3.3 D-Cloud コントローラのインストール手順

#### 3.3.1 OpenStack をインストールした場合

##### 3.3.1.1 deb パッケージからインストールする場合

DS-Bench ターゲット、D-Cloud コントローラの deb パッケージのあるフォルダに移動し以下のコマンドを実行してください。

D-Cloud パッケージをインストールする場合、依存関係により DS-Bench ターゲットパッケージを先にインストールする必要があるのでインストールします。

```
$ sudo dpkg -i -force-depends target.deb
$ sudo apt-get install -f
$ sudo dpkg -i d-cloud-withOpenStack.deb
```

DS-Bench ターゲットパッケージインストール中はすべて未入力 Enter キーを押下します。

D-Cloud インストール実行中に以下の内容を聞かれます。お使いの環境に合わせて設定してください。ここでは D-Cloud コントローラデーモン、D-Cloud リソースデーモンのポート番号はデフォルト値を使用するため、未入力のまま Enter キーを押下します。

```
Input Port Number for D-Cloud Controller Daemon [ 9903 ]:  
Input Port Number for D-Cloud Resource Daemon [ 9904 ]:  
Input Port Number for D-Cloud Map Server Daemon [ 7000 ]: 9905  
Input Path for novarc of OpenStack [ /var/dcloud/openstack/creds/novarc ]: /var/nova/novarc
```

※novarc のパスについては OpenStack インストール時に作成したパスを指定してください。デフォルトでは/home/ユーザー名/novarc

### 3.3.2 OpenStack をインストールしていない場合

#### 3.3.2.1 deb パッケージからインストールする場合

DS-Bench ターゲット、D-Cloud コントローラの deb パッケージのあるフォルダに移動し以下のコマンドを実行してください。

```
$ sudo dpkg -i --force-depends target.deb  
$ sudo apt-get install -f  
$ sudo dpkg -i d-cloud-withoutOpenStack.deb
```

インストール実行中に以下の内容を聞かれます。お使いの環境に合わせて設定してください。

```
Input Port Number for D-Cloud Controller Daemon [ 9903 ]:  
Input Port Number for D-Cloud Resource Daemon [ 9904 ]:  
Input Port Number for D-Cloud Map Server Daemon [ 7000 ]: 9905
```

ここでは D-Cloud コントローラデーモン、D-Cloud リソースデーモンのポート番号はデフォルト値を使用するため、未入力のまま Enter キーを押下します。

以上で D-Cloud のインストール、設定は終了です。

## 4 OpenStack

### 4.1 OpenStack コントローラ+計算ノード構築手順

仮想マシンを使用する場合は D-Cloud コントローラが起動しているサーバ、またはインストール予定のサーバに OpenStack のインストールを行ってください。D-Cloud が使用する OpenStack は通常は D-Cloud コントローラ上のみで動作します。

#### 4.1.1 ネットワークの設定

以下の手順でブリッジの設定を行います。

例としてここでは D-Cloud の IP アドレスを「**192.168.1.2**」ブリッジ名を「**br0**」とします。ネットワークの編集が終わったら必ずネットワークの再起動を行ってください。

```
$ sudo apt-get install -y bridge-utils

$ sudo vi /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet manual
up ifconfig eth0 0.0.0.0 up

auto br0
iface br0 inet static
address 192.168.1.2
network 192.168.1.0
netmask 255.255.255.0
gateway 192.168.1.1
dns-nameservers 192.168.1.1
bridge_ports eth0
bridge_stp off
bridge_maxwait 1

$sudo /etc/init.d/networking restart
```

※**dns-nameservers** は Ubuntu10.04 では必要ありませんが、Ubuntu12.04 では必ず記述してください。

#### 4.1.2 ホスト名の追加

/etc/hosts に OpenStack コントローラ、計算ノードのホスト名を登録します。

例としてコントローラの IP アドレスを「**192.168.1.2**」、ホスト名を「**controller**」を追記します。

※計算ノードの IP アドレスを「**192.168.1.3**」、ホスト名を「**compute**」とします。

```
$ sudo vi /etc/hosts
127.0.0.1    localhost
127.0.1.1    controller

192.168.1.2    controller
192.168.1.3    compute
```

#### 4.1.3 リポジトリの追加

OpenStack は、Ubuntu12.04 ではパッケージとして提供されているのでそのままインストール出来ますが、Ubuntu10.04 でインストールを行う場合はリポジトリを追加してそこからインストールを行ってください。

以下の手順でリポジトリの追加を行います。

```
$ gpg -keyserver keyserver.ubuntu.com -recv 7D21C2EC3D1B4472
$ gpg -export -armor 7D21C2EC3D1B4472 | sudo apt-key add -

$ sudo vi /etc/apt/sources.list
deb http://apt.cloudscaling.com/openstack-release/2011.3/ubuntu lucid main
```

最新のパッケージリストを取得します。

```
$ sudo apt-get update
.
パッケージリストを読み込んでいます... 0%
パッケージリストを読み込んでいます... 100%
パッケージリストを読み込んでいます... 完了
```

#### 4.1.4 OpenStack のインストール

OpenStack のインストールを行います。

```
$ sudo apt-get install -y rabbitmq-server
パッケージリストを読み込んでいます... 0%
.
.
ldconfig deferred processing now taking place
$
```

```
$ sudo apt-get install -y python-greenlet python-mysqldb
パッケージリストを読み込んでいます... 0%
.
.
python-support のトリガを処理しています ...
$
```

```
$ sudo apt-get install -y nova-network nova-compute nova-scheduler
パッケージリストを読み込んでいます... 0%
.
.
ldconfig deferred processing now taking place
$
```

```
$ sudo apt-get install -y libvirt-bin
パッケージリストを読み込んでいます... 0%
.
.
libvirt-bin はすでに最新バージョンです。
libvirt-bin は手動でインストールしたと設定されました。
アップグレード: 0 個、新規インストール: 0 個、削除: 0 個、保留: 110 個。
$
```

```
$ sudo apt-get install -y euca2ools unzip
パッケージリストを読み込んでいます... 0%
.
.
cloud-utils (0.11-0ubuntu1) を設定しています ...
$
```

```
$ sudo apt-get install -y nova-volume nova-vncproxy nova-api nova-ajax-console-proxy
パッケージリストを読み込んでいます... 0%
.
.
python-support のトリガを処理しています ...
$
```

```
$ sudo apt-get install -y glance
パッケージリストを読み込んでいます... 0%
.
.
$
```

Ubuntu12.04 で設定を行っている場合は以下のパッケージもインストールしてください。

```
$ sudo apt-get install -y nova-cert
パッケージリストを読み込んでいます... 0%
.
.
$
```

mysql の設定スクリプトを編集します。赤字部分の `MYSQL_PASS` には root ユーザのパスワードを、`NOVA_PASS` の部分には nova ユーザのパスワードを任意で設定してください。

ここではそれぞれ「**nova**」で設定を行っています。

お使いの環境に合わせて Ubuntu10.04、Ubuntu12.04 の設定を参照してください。

・ Ubuntu10.04 の場合

```
$ sudo vi ~/nova-mysql.sh
#!/bin/bash

MYSQL_PASS=nova
NOVA_PASS=nova

cat << MYSQL_PRESEED | sudo debconf-set-selections
mysql-server-5.1 mysql-server/root_password password $MYSQL_PASS
mysql-server-5.1 mysql-server/root_password_again password $MYSQL_PASS
mysql-server-5.1 mysql-server/start_on_boot boolean true
MYSQL_PRESEED

sudo apt-get install -y mysql-server
sudo sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/my.cnf
sudo service mysql restart
sudo mysql -u root -p$MYSQL_PASS -e 'CREATE DATABASE nova;'
sudo mysql -u root -p$MYSQL_PASS -e "GRANT ALL PRIVILEGES ON *.* TO 'nova'@'%' WITH GRANT OPTION;"
sudo mysql -u root -p$MYSQL_PASS -e "SET PASSWORD FOR 'nova'@'%' = PASSWORD('$NOVA_PASS');"
```

・ Ubuntu12.04 の場合

```
$ sudo vi ~/nova-mysql.sh
#!/bin/bash

MYSQL_PASS=nova
NOVA_PASS=nova

cat << MYSQL_PRESEED | sudo debconf-set-selections
mysql-server-5.5 mysql-server/root_password password $MYSQL_PASS
mysql-server-5.5 mysql-server/root_password_again password $MYSQL_PASS
mysql-server-5.5 mysql-server/start_on_boot boolean true
MYSQL_PRESEED

sudo apt-get install -y mysql-server
sudo sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/my.cnf
sudo service mysql restart
sudo mysql -u root -p$MYSQL_PASS -e 'CREATE DATABASE nova;'
sudo mysql -u root -p$MYSQL_PASS -e "GRANT ALL PRIVILEGES ON *.* TO 'nova'@'%' WITH GRANT OPTION;"
sudo mysql -u root -p$MYSQL_PASS -e "SET PASSWORD FOR 'nova'@'%' = PASSWORD('$NOVA_PASS');"
sudo mysql -u root -p$MYSQL_PASS -e "GRANT ALL PRIVILEGES ON nova.* TO nova@localhost IDENTIFIED BY '$NOVA_PASS';"
```

編集したスクリプトを実行します。

```
$ sudo bash ~/nova-mysql.sh
パッケージリストを読み込んでいます... 0%
.
.
mysql start/running, process 11521
$
```

OpenStack 関連の権限を実行します。Ubuntu12.04 では赤字も表示されます。

```
$ sudo chown -R nova:root /etc/nova
$
$ sudo ls -ltrR /etc/nova
/etc/nova:
合計 12
-rw-r--r-- 1 nova root 19 2011-09-23 03:25 nova-compute.conf
-rw-r--r-- 1 nova root 3080 2011-09-23 02:53 api-paste.ini
-rw----- 1 nova root 254 2011-09-23 02:53 nova.conf
-rw----- 1 nova root 254 2012-08-25 02:53 policy.json
$
```

OpenStack の設定ファイルへ追記します。赤字部分は任意で設定してください。  
お使いの環境に合わせて Ubuntu10.04、Ubuntu12.04 の設定を参照してください。

・ Ubuntu10.04 の場合

```
$ sudo vi /etc/nova/nova.conf

--dhcpbridge_flagfile=/etc/nova/nova.conf
--dhcpbridge=/usr/bin/nova-dhcpbridge
--logdir=/var/log/nova
--state_path=/var/lib/nova
--lock_path=/var/lock/nova
--flagfile=/etc/nova/nova-compute.conf
--force_dhcp_release=True
--use_deprecated_auth
--verbose
--sql_connection=mysql://nova:nova@controller/nova
--network_manager=nova.network.manager.FlatDHCPManager
--public_interface=eth0
--flat_interface=eth0
--flat_injected=false
--bridge=eth0
--bridge_interface=br100
--network_host=192.168.1.2
--routing_source_ip=192.168.1.2
--fixed_range=10.0.0.0/8
--network_size=1024
--cc_host= 192.168.1.2
--ec2_url=http://192.168.1.2:8773/services/Cloud
--s3_host=192.168.1.2
--s3_dmz=192.168.1.2
--glance_api_servers=192.168.1.2:9292
--image_server=nova.image.glance.GlanceImageService
--rabbit_host=192.168.1.2
--vnc_enabled=true
--vncproxy_url=http://192.168.1.2:6080
--vnc_console_proxy_url=http://192.168.1.2:6080
```

※nova:nova@controller の設定は nova:[mysql の nova ユーザのパスワード]@[コンピュータ名] を指定します。

## • Ubuntu12.04 の場合

```
$ sudo vi /etc/nova/nova.conf

--dhcpbridge_flagfile=/etc/nova/nova.conf
--dhcpbridge=/usr/bin/nova-dhcpbridge
--logdir=/var/log/nova
--state_path=/var/lib/nova
--lock_path=/var/lock/nova
--flagfile=/etc/nova/nova-compute.conf
--force_dhcp_release=True
--use_deprecated_auth
--verbose
--sql_connection=mysql://nova:nova@controller/nova
--network_manager=nova.network.manager.FlatDHCPManager
--public_interface=eth0
--flat_interface=eth0
--flat_injected=false
--bridge=eth0
--bridge_interface=br100
--network_host=192.168.1.2
--routing_source_ip=192.168.1.2
--fixed_range=10.0.0/8
--network_size=1024
--cc_host= 192.168.1.2
--ec2_url=http://192.168.1.2:8773/services/Cloud
--s3_host=192.168.1.2
--s3_dmz=192.168.1.2
--glance_api_servers=192.168.1.2:9292
--image_server=nova.image.glance.GlanceImageService
--rabbit_host=192.168.1.2
--vnc_enabled=true
--vncproxy_url=http://192.168.1.2:6080
--vnc_console_proxy_url=http://192.168.1.2:6080
--iscsi_helper=tgtadm
--libvirt_use_virtio_for_bridges
--connection_type=libvirt
--root_helper=sudo nova-rootwrap
```

プロセスを起動します。

```
$ sudo nova-manage db sync
$
```

※Ubuntu12.04 の場合、上記コマンド実行時に DEBUG 用のメッセージが出力される場合がありますがそのまま設定を続けてください。

```
$ sudo restart nova-network
nova-network start/running, process 11642
$
```

```
$ sudo restart nova-api
nova-api start/running, process 11695
$
```

```
$ sudo restart nova-compute
nova-compute start/running, process 11710
$
```

```
$ sudo restart nova-scheduler
nova-scheduler start/running, process 11780
$
```

```
$ sudo restart glance-api
glance-api start/running, process 11876
$
```

```
$ sudo restart glance-registry
glance-registry start/running, process 11883
$
```

Ubuntu12.04 の場合は以下も実行してください。

```
$ sudo restart nova-cert
nova-scheduler start/running, process 11751
$
```

#### 4.1.5 D-Cloud 用のプロジェクト作成

管理者を作成します。ここでは「**nova-admin**」を作成します。

```
$ sudo nova-manage user admin nova-admin
.
.
$
```

プロジェクトを作成します。ここでは「**d-cloud**」を作成します。

```
$ sudo nova-manage project create d-cloud nova-admin
$
```

ネットワークの設定をします。赤字部分は任意で設定してください。

```
$ sudo nova-manage network create public --bridge=br100 --bridge_interface=eth0
--fixed_range_v4=10.0.1.0/24 --num_network=1 --network_size=256
$
```

ネットワーク設定の確認をします

```
$ sudo nova-manage network list
id      IPv4      IPv6      start address      DNS1      DNS2      VlanID      project      uuid
1       10.0.1.0/24 None 10.0.1.2      8.8.4.4   None     None        None         None
$
```

環境設定ファイルを展開します。展開するパスは任意で設定してください。

```
$ sudo nova-manage project zipfile d-cloud nova-admin ./novacred.zip
$
$ unzip novacred.zip
Archive:  novacred.zip
extracting: novarc
extracting: pk.pem
extracting: cert.pem
extracting: cacert.pem
$
$ source novarc
$
```

仮想マシンで使用するポートを開放します。ここでは DS-Bench ターゲットデーモンのポート番号をデフォルト値の「**9804**」として記述しています。お使いの環境に合わせて設定してください。

```
$ euca-authorize -P icmp -t -1:-1 default
$
$ euca-authorize -P tcp -p 22 default
$
$ euca-authorize -P tcp -p 9804 default
$
$ euca-authorize -P tcp -p 5001 default
$
```

## 4.2 OpenStack 計算ノード構築手順

計算ノード必要な場合は以下の手順で必要に応じてインストールおよび設定してください。計算ノードを追加すると計算ノードの計算資源を利用して新規に仮想マシンを作成できます。

### 4.2.1 ホスト名の追加

/etc/hosts に OpenStack コントローラ、計算ノードのホスト名を登録します。

例としてコントローラの IP アドレスを「**192.168.1.2**」、ホスト名を「**controller**」を追記します。

※計算ノードの IP アドレスを「**192.168.1.3**」、ホスト名を「**compute**」とします。

```
$ sudo vi /etc/hosts
127.0.0.1    localhost
127.0.1.1    compute

192.168.1.2    controller
192.168.1.3    compute
```

### 4.2.2 リポジトリの追加

手順については「[4.1.3 リポジトリの追加](#)」を参照してください。

### 4.2.3 OpenStack(計算ノード)のインストール

計算ノードに必要なパッケージのみをインストールします。

```
$ sudo apt-get install -y nova-compute
パッケージリストを読み込んでいます... 0%
.
.
python-support のトリガを処理しています ...
$
```

OpenStack 関連の権限を変更します

```
$ sudo chown -R nova:root /etc/nova
$
$ ls -ltrR /etc/nova
/etc/nova:
合計 12
-rw-r--r-- 1 nova root   19 2011-09-23 03:25 nova-compute.conf
-rw-r--r-- 1 nova root 3080 2011-09-23 02:53 api-paste.ini
-rw----- 1 nova root  254 2011-09-23 02:53 nova.conf
$
```

OpenStack の設定ファイル (/etc/nova/nova.conf) を修正します。

ファイル内容は「[4.1.4 OpenStack のインストール](#)」で修正したファイルと同じ内容です。

※ファイルをコピーしても構いません。

プロセスを起動します。

```
$ sudo restart nova-compute
nova-compute start/running, process 11710
$
```

計算ノードとして追加されているか確認します。

```
$ sudo nova-manage service list
Binary          Host              Zone      Status  State  Updated_At
nova-network    controller        nova      enabled :-)    2012-03-06 05:26:37
nova-compute     controller        nova      enabled :-)    2012-03-06 05:26:40
nova-scheduler  controller        nova      enabled :-)    2012-03-06 05:26:40
nova-compute     compute          nova      enabled :-)    None
$
```

※Ubuntu12.04 の場合は nova-cert も表示されます。

## 5 DS-Bench

### 5.1 Ubuntu12.04 への対応

対応方法は「[3.1 Ubuntu12.04 への対応](#)」を参照ください。

### 5.2 コントローラのインストール手順

#### 5.2.1 CPAN シェルの設定

この操作は、インストール手順の中で CPAN モジュール (Sedna) をインストールするために行うものです。したがって、ここでモジュールをインストールする必要はありません。また、**これまでに一度でも CPAN シェルを使って Perl モジュールをインストールしたことがある場合、この項の操作は必要ありません。**

初めて CPAN を利用する場合、以下のコマンドを実行してください。

```
$ sudo perl -MCPAN -e shell
```

実行すると、CPAN を利用するための設定の入力を促されます。お使いの環境に応じて値を入力してください。

設定が終わると、プロンプトが返ってきます。問題なければ”o conf commit”で設定を保存してください。その後、”q”で CPAN シェルを終了します。

```
cpan[1]> o conf commit  
cpan[2]> q
```

#### 5.2.2 DS-Bench コントローラのインストール

##### 5.2.2.1 deb パッケージからインストール

DS-Bench コントローラの deb パッケージのあるフォルダに移動し、以下のコマンドを実行してください。

```
$ sudo dpkg -i --force-depends ctrl.deb  
$ sudo apt-get install -f
```

インストール中、Apache の DocumentRoot のパスを聞かれます。すでに Apache をお使いの場合、Port:80 の DocumentRoot の値を指定してください。

ポート番号、IP アドレス等はお使いの環境に合わせて設定してください。ここではポート番号、DB 名はデフォルト値を使用するため、未入力のまま Enter キーを押下します。

ここでは例として D-Cloud コントローラの IP アドレスを 192.168.1.2 とします。

```
Input DSBench Document Root [ /var/www/dsbench ]: [DocumentRoot のパス]
Input Port Number for DSBench-Controller Daemon [ 9803 ]:
Input Port Number for DSBench-Target Daemon [ 9804 ]:
Input Hostname or IP Address for DCloud [ 127.0.0.1 ]: 192.168.1.2
Input Port Number for DCloud [ 9903 ]:
Input Hostname or IP Address for Sedna DB [ 127.0.0.1 ]:
Input DB Name for DSBench [ dsbench ]:
Input Benchmark Collection Name for DSBench [ result_benchmark ]:
Input Benchmark Scenario Collection Name for DSBench [ result_benchmark_set ]:
```

### 5.2.3 Apache の設定

本システムのために Apache のチューニングが必要になることがあります。なお Apache がインストールされていないマシンで設定を行っている場合は、controller の依存パッケージとしてインストールされます。

/etc/apache2/sites-available/default の DocumentRoot の値を\${DOCUMENTROOT}に設定し、以下の内容を追記してください。ここで、\${DOCUMENTROOT}はインストールの際に指定した DocumentRoot のパスを表します。バッティングする Directory ブロックがすでに記述されている場合、念のため削除ないしコメントアウトすることをお勧めします。

```
<Directory ${DocumentRoot}>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

<Directory "${DocumentRoot}/cgi-bin">
    DirectoryIndex main.cgi
    AllowOverride None
    AddHandler cgi-script .cgi .pl
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
</Directory>
```

設定が終わった後、**必ず Apache を再起動してください**。また、本システムは CGI を使っていますが ScriptAlias を必要としません。該当ディレクトリの ScriptAlias は外してください。

#### 5.2.4 Sedna DB サーバの設定(任意)

Sedna DB サーバの IP アドレス等の設定を変更することができます。後で DS-Bench 画面からも設定することができます。

/var/dsbench/config/sedna.conf ファイルを編集します。編集例を以下に示します。

```
SEDNA_HOST      : 192.168.1.4
SEDNA_DB_NAME   : dsbench
SEDNA_DB_USERNAME : SYSTEM
SEDNA_DB_PASSWORD : MANAGER
SEDNA_COLLECTION_RESULT_BENCHMARK : result_benchmark
SEDNA_COLLECTION_RESULT_BENCHMARK_SET : result_benchmark_set
```

#### 5.2.5 D-Cloud コントローラの設定(任意)

D-Cloud コントローラの IP アドレス等の設定を変更することができます。後で DS-Bench 画面からも設定することができます。

/var/dsbench/config/cloud.conf ファイルを編集します。編集例を以下に示します。

```
CLOUD_CONTROLLER : 192.168.1.2
CLOUD_PORT       : 9903
```

## 5.3 SednaDB サーバのインストール

### 5.3.1 deb パッケージからインストール

Sedna の deb パッケージのあるフォルダに移動し、以下のコマンドを実行してください。

```
$ sudo dpkg -i Sedna.deb
```

インストール中に DB 名等を聞かれます。環境に合わせて設定してください。  
ここではデフォルト値を使用するため、未入力のまま Enter キーを押下します。

```
Input DB Name for DSbench [ dsbench ]:  
Input Benchmark Collection Name for DSbench [ result_benchmark ]:
```

### 5.3.2 Sedna の修正

下記 URL から Sedna-0.002.tar.gz をダウンロードして展開してください。

<http://www.cpan.org/authors/id/D/DR/DRUOSO/>

展開したフォルダ上にある Sedna.xs ファイルの 353 行目を以下の通りに変更します。

```
(変更前)memcpy((char*)((int)result + curlen), buffer, ret);  
(変更後)memcpy((result + curlen), buffer, ret);
```

変更ができれば保存してコンパイルをおこない、インストールをしてください。

```
$perl Makefile.PL  
$make  
$make test  
$sudo make install
```

## 5.4 DS-Bench ターゲットのインストール

### 5.4.1 deb パッケージからインストール

Sedna の deb パッケージのあるフォルダに移動し、以下のコマンドを実行してください。

```
$ sudo dpkg -i --force-depens target.deb  
$ sudo apt-get install -f
```

インストール実行中は以下の内容を聞かれるため、環境に合わせて入力してください。  
ここではデフォルト値を使用するため、未入力のまま Enter キーを押下します。

```
Input Port Number for DSbench-Controller Daemon [ 9803 ]:  
Input Port Number for DSbench-Target Daemon [ 9804 ]:
```

### 5.4.2 リポジトリファイルの修正

ベンチマークプログラムをインストールするために、リポジトリファイルへ追記します。

DS-Bench コントローラをインストールする際にリポジトリを作成していますので、参照先を DS-Bench コントローラとします。追記例を以下に示します。

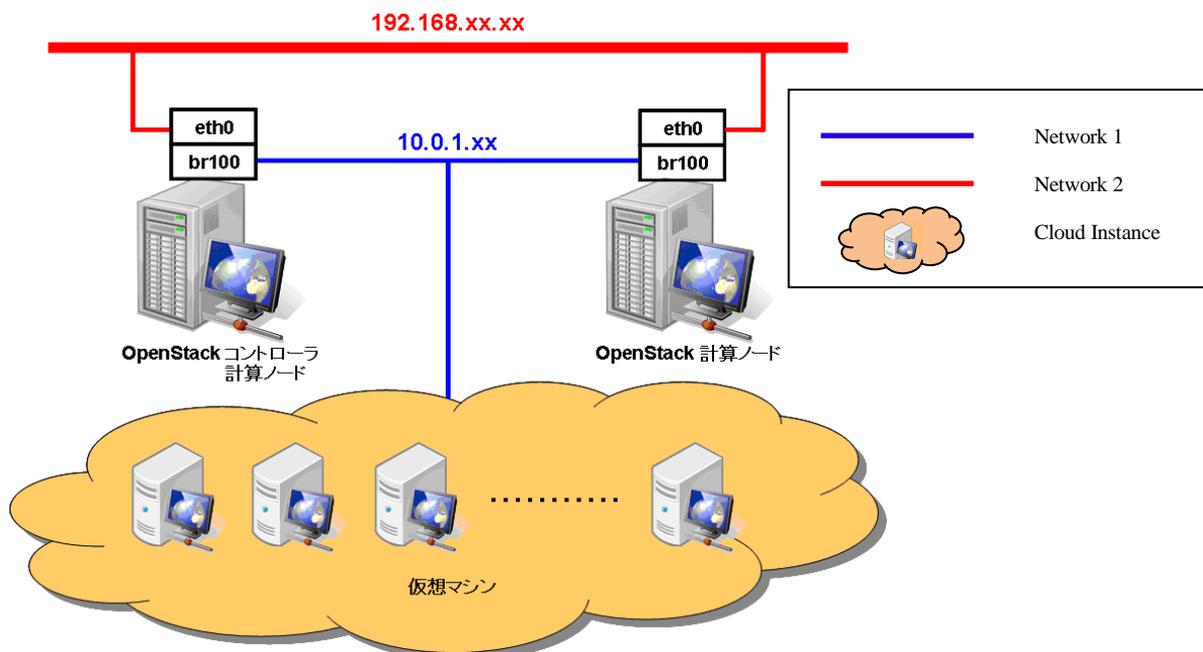
```
$ cat /etc/apt/sources.list.d/dsb.list  
deb http://[コントローラの IP or ホスト名]/deb/ ./  
deb-src http:// [コントローラの IP or ホスト名]/deb/ ./
```

リポジトリファイル修正後、**必ずパッケージリストを更新してください。**

```
$ sudo apt-get update
```

## 6 仮想マシン

### 6.1 仮想ターゲット使用時のネットワーク設定



DS-Bench コントローラ (IP アドレス 192.168.X.X) と仮想マシン (IP アドレス 10.0.1.X) 間で通信を行えるように、DS-Bench コントローラのルーティングテーブルの設定を行う必要があります。

※DS-Bench 物理ターゲットと仮想ターゲット間で通信を行う場合は物理ターゲット上でもルーティングテーブル設定を行ってください。

経路情報を/etc/network/interface に追記します。

```
$ sudo vi /etc/network/interface  
  
up route add -net 10.0.1.0 netmask 255.255.255.0 gw D-CloudIP アドレス
```

設定後、ネットワークの再起動を行い、route コマンドで設定が反映されていることを確認し、ping が通ることを確認してください。

## 6.2 仮想ターゲットイメージ作成手順

### 6.2.1 初期イメージの登録

Ubuntu 公式 HP 等からダウンロードしたイメージをコントローラサーバへアップロードします。

<http://uec-images.ubuntu.com/>

イメージファイルが格納されているディレクトリへ移動します。

```
$ cd /var/nova/images/test/
$ pwd
/var/nova/images/test/
```

お使いの環境に合わせて Ubuntu10.04、Ubuntu12.04 の設定を参照してください。

Ubuntu10.04 の場合

- イメージファイル `/var/nova/images/test/lucid-server-cloudimg-amd64.img`
- カーネルファイル `/var/nova/images/test/lucid-server-cloudimg-amd64-vmlinuz-virtual`

OpenStack へ仮想カーネルを登録します。ここでは登録名を「**org-kernel**」とします。

```
$ sudo nova-manage image kernel_register lucid-server-cloudimg-amd64-vmlinuz-virtual
admin --name=org-kernel
Image registered to 1 (00000001).
```

イメージを OpenStack へ登録します。ここでは登録名を「**org-img**」とします。

カーネルは先ほど登録したものを指定します。

```
$ sudo nova-manage image image_register lucid-server-cloudimg-amd64.img admin
--name=org-img --kernel=00000001 --cont_format=ami --disk_format=ami
```

イメージが登録されていることを確認します。

```
$ glance index
```

ID	Name	Disk Format	Container	Format	Size
2	org-img	ami	ami		1476395008
1	org-kernel	aki	aki		4108960

Ubuntu12.04 の場合

- イメージファイル `/var/nova/images/test/precise-server-cloudimg-amd64-disk1.img`

OpenStack へ仮想イメージを登録します。ここでは登録名を「`ubuntu-12.04`」とします。

```
$ glance add name="ubuntu-12.04" is_public=true container_format=ovf disk_format=qcow2
< precise-server-cloudimg-amd64-disk1.img
  Uploading image 'ubuntu-12.04'
  ===== [100%] 31.3M/s, ETA 0h 0m 0s
```

仮想イメージが登録されていることを確認します。

```
$ glance index
ID                Name                Disk Format  Container Format  Size
-----
4ec748ac-8e05...  ubuntu-12.04       qcow2       ovf                233832448
```

### 6.2.2 初期イメージの起動

初期イメージを起動します。

まず ssh でログインできるようにするため、鍵を作成します。

作成するディレクトリは「`~/.ssh/`」です。

```
$ euca-add-keypair orgkey > ~/.ssh/orgkey
$
```

作成した鍵のパーミッションを変更します。

```
$ chmod 600 ~/.ssh/orgkey
$
```

初期イメージのイメージ ID を確認します。

```
$source novarc
$ euca-describe-images
IMAGE  ami-00000001  None (ubuntu-12.04)  available  public  machine  instance-store
```

作成した鍵を使用して初期イメージのインスタンスを起動します。

```
$ euca-run-instances -t m1.tiny -k orgkey ami-00000001
RESERVATION    r-i40e30j5    [ホスト名] default
INSTANCE       i-00000001    ami-00000001    server-1 server-1 pending
                orgkey ([ホスト名], None) 0
```

インスタンスの起動確認をします。起動完了後は IP アドレスが振られ、ステータスが「**running**」となります。

```
$ euca-describe-instances
RESERVATION    r-931xv678    d-cloud default
INSTANCE       i-00000001    ami-00000002    10.0.1.2  10.0.1.2  running orgkey ([ホスト
名], [起動サーバ名]) 0    m1.tiny [起動日時]    nova    aki-00000001    ami-00000000
```

### 6.2.3 仮想ターゲットイメージの作成

初期イメージのインスタンスに ssh でログインします。「3.2.初期イメージの起動」で作成した鍵を使用します。ユーザ名は「**ubuntu**」です。

```
$ ssh -i ~/.ssh/orgkey ubuntu@10.0.1.2
The authenticity of host '10.0.1.2 (10.0.1.2)' can't be established.
RSA key fingerprint is 94:f5:22:ef:a0:15:ad:1a:d6:a0:43:b7:8d...
Are you sure you want to continue connecting (yes/no)? yes
.
.
.
ubuntu@server-1:~$
```

DS-Bench ターゲットをインストールします。

インストール方法については、deb パッケージを scp コマンドで起動したインスタンスにコピーを行い、「[5.4 DS-Bench ターゲットのインストール](#)」の手順を参照してください。

```
$ sudo scp -i ~/.ssh/orgkey パッケージ ubuntu@10.0.1.2:/home/ubuntu/
```

リポジトリファイルについては、仮想マシン起動時に初期化されてしまうため、別途作業が必要になります。手順は以下の通りです。

使用するリポジトリファイルをホームディレクトリに作成します。

```
ubuntu@server-1:~$ vi ~/sources.list
```

リポジトリファイルの内容は以下になります。

```
deb http://[コントローラの IP or ホスト名]/deb/ ./  
deb-src http:// [コントローラの IP or ホスト名]/deb/ ./
```

作成したリポジトリファイルを確認します。

```
$ cat ~/sources.list  
deb http://[コントローラの IP or ホスト名]/deb/ ./  
deb-src http:// [コントローラの IP or ホスト名]/deb/ ./
```

起動時に作成したリポジトリファイルが適用されるようにするため、起動スクリプトへ元のリポジトリファイルを上書きコピーする手順を追記します。

```
$ vi /etc/rc.local  
#!/bin/sh -e  
sleep 5  
logger -t initsourceslist cp /home/ubuntu/sources.list /etc/apt/sources.list  
cp /home/ubuntu/sources.list /etc/apt/sources.list  
exit 0
```

起動スクリプトの実行順を変更します。

```
$ sudo mv /etc/rc2.d/S99grub-common /etc/rc2.d/S97grub-common
$
$ sudo mv /etc/rc2.d/S99ondemand /etc/rc2.d/S97ondemand
$
$ sudo mv /etc/rc2.d/S99rc.local /etc/rc2.d/S98rc.local
$
$ sudo mv /etc/rc2.d/S20ds-target /etc/rc2.d/S99ds-target
```

仮想マシンをシャットダウンします。

```
ubuntu@server-1:~$ sudo shutdown -h now
Broadcast message from ubuntu@server-1
      (/dev/pts/0) at ...
.
.
$
```

シャットダウンしたインスタンスからイメージを作成します。インスタンスのディレクトリを確認します。

```
$ ls -l /var/lib/nova/instances/
total 8
drwxr-xr-x 2 nova nogroup 4096 yyyy-mm-dd hh:mi _base
drwxr-xr-x 2 nova nogroup 4096 yyyy-mm-dd hh:mi instance-00000001
$
```

イメージを作成します。ここでは、作成するイメージ名を「**target-img**」、保存ディレクトリを「**/var/nova/images/test**」とします。

```
$ qemu-img convert -f qcow2 /var/lib/nova/instances/instance-00000001/disk -o  
raw /var/nova/images/test/target-img
```

作成したイメージを確認します。

・ Ubuntu10.04

```
$ ls -l /var/nova/images/test/  
total 1991356  
-rw-r--r-- 1 root root 1476395008 yyyy-mm-dd hh:mi lucid-server-cloudimg-amd64.img  
-rw-rw-rw- 1 root root 4110432 yyyy-mm-dd hh:mi lucid-server-cloudimg-amd64-vmlinuz-virtual  
-rw-r--r-- 1 root root 1476395008 yyyy-mm-dd hh:mi target-img  
$
```

・ Ubuntu12.04

```
$ ls -l /var/nova/images/test/  
total 1991356  
-rw-r--r-- 1 root root 1476395008 yyyy-mm-dd hh:mi precise-server-cloudimg-amd64-desk1.img  
-rw-r--r-- 1 root root 1476395008 yyyy-mm-dd hh:mi target-img  
$
```

## 6.2.4 仮想ターゲットマシンの登録

「[6.1.3 仮想ターゲットイメージ作成手順](#)」で作成したターゲットイメージを OpenStack へ登録します。

イメージファイルが格納されているディレクトリへ移動します。

```
$ cd /var/nova/images/test/
$ pwd
/var/nova/images/test/
```

仮想ターゲットイメージを登録します。

カーネルは初期カーネルを使用します。

- Ubuntu10.04

```
$ sudo nova-manage image image_register target-img admin --name=target-img --kernel=00000001
--cont_format=ami --disk_format=ami
Image registered to 3 (00000003).
IMAGE aki-00000001    local (testkernel)    available    public        x86_64    kernel
```

インスタンスが登録されていることを確認します。

```
$ glance index
```

ID	Name	Disk Format	Container Format	Size
3	target-img	ami	ami	1476395008
2	org-img	ami	ami	1476395008
1	org-kernel	aki	aki	4108960

- Ubuntu12.04

```
$ glance add name="12.04-target" is_public=true container_format=ovf disk_format=qcow2 <
target-img
  Uploading image '12.04-target'
===== [100%] 31.3M/s, ETA 0h 0m 0s
Added new image with ID: 4ec748ac-8w35...    public        x86_64    kernel
```

インスタンスが登録されていることを確認します。

```
$ glance index
```

ID	Name	Disk Format	Container Format	Size
4ec748ac-8w35-40b6-8de6-...	12.04-target	qcow2	ovf	2147483648
4ec748ac-8e05-40b6-8de6-...	ubuntu-12.04	qcow2	ovf	233832448

登録後、必ず **D-Cloud** のデーモンを再起動してください。

```
$ sudo /etc/init.d/dc-mapserver restart
Stopping ... success
Starting ... success
$
$ sudo /etc/init.d/dc-resource-dsb restart
Stopping ... success
Starting ... success
$
$ sudo /etc/init.d/dcld restart
Stopping ... success
Starting ... success
$
```

DS-Bench 画面にて、仮想ターゲットマシンが登録されていることを確認します。

[main 画面](#) > [Configuration 画面](#) > [Target list](#)

Target list			
No	Machine name	Type	Status
1	trg-tutorial	physical	Active
2	target-img-3-m1.tiny.00	virtual	Power off
3	target-img-3-m1.tiny.01	virtual	Power off
4	target-img-3-m1.tiny.02	virtual	Power off

D-Cloud の設定ファイル値 (/var/dcloud/config/resource\_dsb.xml の<instances><limit>の値) が「3」となっているため、表示されるイメージ数が「3」となっています。

上記設定値は仮想マシンの起動上限値のため、変更する際は注意してください。



## DEOS プロジェクト